



The Separatrix Algorithm for Synthesis and Analysis of Stochastic Simulations with Applications in Disease Modeling

Daniel J. Klein^{1*}, Michael Baym^{1,2,3}, Philip Eckhoff¹

1 Institute for Disease Modeling, Bellevue, Washington, United States of America, **2** Department of Systems Biology, Harvard Medical School, Boston, Massachusetts, United States of America, **3** Department of Mathematics, Massachusetts Institute of Technology, Boston, Massachusetts, United States of America

Abstract

Decision makers in epidemiology and other disciplines are faced with the daunting challenge of designing interventions that will be successful with high probability and robust against a multitude of uncertainties. To facilitate the decision making process in the context of a goal-oriented objective (e.g., eradicate polio by 2018), stochastic models can be used to map the probability of achieving the goal as a function of parameters. Each run of a stochastic model can be viewed as a Bernoulli trial in which “success” is returned if and only if the goal is achieved in simulation. However, each run can take a significant amount of time to complete, and many replicates are required to characterize each point in parameter space, so specialized algorithms are required to locate desirable interventions. To address this need, we present the Separatrix Algorithm, which strategically locates parameter combinations that are expected to achieve the goal with a user-specified probability of success (e.g. 95%). Technically, the algorithm iteratively combines density-corrected binary kernel regression with a novel information-gathering experiment design to produce results that are asymptotically correct and work well in practice. The Separatrix Algorithm is demonstrated on several test problems, and on a detailed individual-based simulation of malaria.

Citation: Klein DJ, Baym M, Eckhoff P (2014) The Separatrix Algorithm for Synthesis and Analysis of Stochastic Simulations with Applications in Disease Modeling. PLoS ONE 9(7): e103467. doi:10.1371/journal.pone.0103467

Editor: Guido Germano, University College London, United Kingdom

Received: August 16, 2013; **Accepted:** July 3, 2014; **Published:** July 31, 2014

Copyright: © 2014 Klein et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Funding: This work was supported by Bill and Melinda Gates through the Global Good Fund at Intellectual Ventures Laboratory. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* Email: dklein@intven.com

Introduction

Decision makers are tasked with the challenging job of determining how best to achieve specific program goals despite large amounts of uncertainty. In public health, for example, policy makers must decide which interventions to use and when, as well as the demographic sub-population to target, so as to achieve programmatic goals. Global eradication of a disease is a particularly relevant example of one such goal. Malaria is the target of a Global Eradication Campaign announced in 2007, and a successful campaign will most likely require combining several different interventions, with the particular combination tailored to the local transmission setting [1]. Similarly, poliomyelitis has been the target of a global eradication initiative since 1988 [2], wherein oral [3] and inactivated [4] polio vaccines are distributed through various routine and supplementary immunization activities [5].

To achieve these goals with high probability, complex interactions between decisions (e.g. deploy bednets to 80% of the population) and mechanistic uncertainties (e.g. the extent to which mosquitoes feed indoors) must be considered systematically. Computer models enable these interactions to be simulated thousands of times *in silico* before trying in the real world. When using a model to evaluate an objective, each stochastic run can be viewed as a Bernoulli trial in which “success” is returned if and only if the programmatic goals are achieved in simulation. Many

trials (simulations) are required to adequately characterize the underlying probability of success for each parameter configuration. Mapping the success probability across parameter space is highly desired, however the operational regime consists of a stochastic model that can take hours to produce each binary outcome. Efficient algorithms are thus required to best use limited computational resources, particularly now that increasingly-detailed computational models are being used directly in guiding policy decisions in HIV [6, 7], and in campaign design for influenza [8–11], polio [12], and malaria [13–17].

Fortunately, many campaigns do not require achieving full population coverage in order to succeed in interrupting transmission, with the required level depending on the disease’s basic reproductive number. Combined with the phenomenon that increasing coverage becomes increasingly expensive for marginal benefits, threshold phenomenon suggest that the probability success can saturate close to 100% while campaign costs continue to rise with coverage. Thus, regions of campaign and parameter space in which success is almost guaranteed are not particularly interesting. Similarly, regions in which the probability of success is low are uninteresting. The key algorithmic challenge lies in concentrating computational resources in regions of parameter space in which the success probability is near an intermediate target value, such as 80% or 95%.

Related work has primarily focused on analytical techniques for deterministic simulation models. For simulations of this type, the only source of output variation comes from (epistemic) uncertainty about parameter values. Epistemic variation is classically analyzed for uncertainty and sensitivity using numerical sampling methods such as Latin Hypercube Sampling (LHS), see [18,19]. Stochastic models, in which aleatory uncertainty stemming from randomized components is included, are more difficult to analyze because each parameter configuration maps to a multitude of outcomes. Nonetheless, some of the techniques from deterministic model analysis have been adapted to work with stochastic simulation models by running the model many times for each parameter configuration, and using the average as if it were deterministic [20–25]. A second approach to the analysis of stochastic models is the response surface methodology, see [26–30]. The main idea is to fit an easy-to-evaluate metamodel [31] to the mean of a collection of simulation runs for analysis [32] or optimization [33]. While typically based on polynomials, numerous metamodels have been explored including radial basis functions [34], neural networks [35], thin-plate splines [36], support vector machines, and Gaussian processes (kriging) [37,38].

Another area of related work to consider is experiment design, the problem of choosing which simulations to run on the next iteration. Simple designs include factorial and fractional factorial designs [39] that improve upon varying one parameter at a time by exploring locally extreme variations. Distance maximizing and information theoretic approaches have been explored [40], with much of the work directed towards designing computer experiments [41–44]. More advanced and computationally intensive approaches to experiment design include a class of algorithms that seek to optimize the expected information returned by the samples by minimizing a loss function [45,46] or maximizing the expected information gain [47–49].

None of the above work satisfactorily addresses the needs of decision makers, who need efficient algorithms to draw meaningful conclusions from detailed stochastic simulations with respect to a goal-oriented objective. To address these needs, we present the Separatrix Algorithm, which directs computational resources towards identifying and resolving combinations of policies and uncertain model parameters that will achieve the goal successfully with a given probability. The algorithm was named after a mathematical term, separatrix, representing a boundary separating behavioral modes. Here, we use the term as a synonym for the desired isocline of the underlying success probability function. While in this work we present the Separatrix Algorithm in the context of computational epidemiology, the methods extend directly to other computational disciplines and further to the scientific domain, wherein laboratory experimentation replaces computer simulation.

The main contributions of this work include (1) an iterative approach to stochastic model evaluation in the context of a goal-oriented objective that consists primarily of (2) a consistent kernel-based regression algorithm that estimates the full distribution of the probability of achieving the goal at a point based on observed simulation outcomes, and (3) a design-of-experiments algorithm that places the next M sample points so as to strategically gain information about the separatrix. These elements are illustrated in Figure 1.

Methods

The Separatrix Algorithm combines binomial regression, based on novel kernel methods, with an experiment design procedure, called igBDOE, that maximizes expected information gain. The

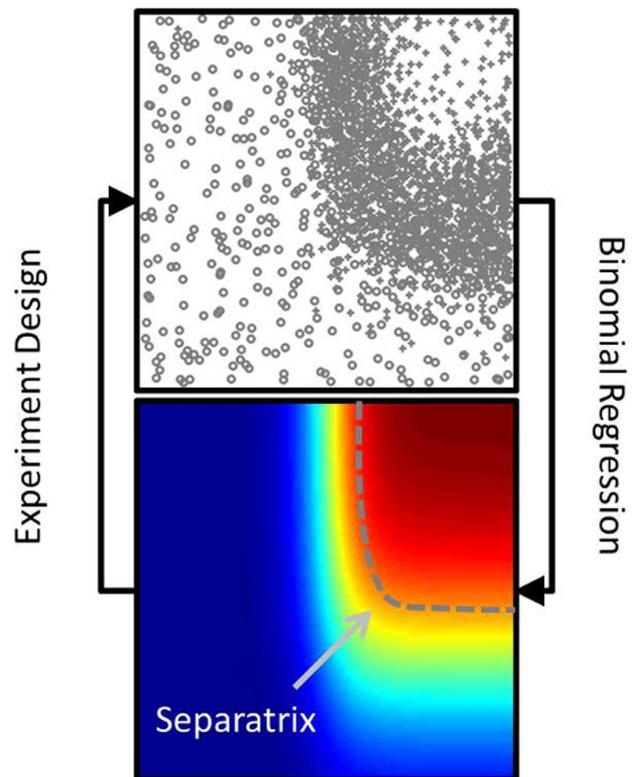


Figure 1. The Separatrix Algorithm addresses two main sub-problems. The first is to use observed binary outcomes (top) to estimate the probability of success (bottom), and the second is to choose new points to sample. This is done so as to identify a particular isocline, called the separatrix, as illustrated by the dashed gray line. doi:10.1371/journal.pone.0103467.g001

algorithm progresses by iteratively choosing new parameter configurations to simulate (i.e. sample points) from a D -dimensional box, called the *parameter space* or *sample space*, that we will denote by \mathcal{B} .

Simulations that are complete will be denoted by $X_i \in \mathcal{B}$ and $Y_i \in \{0,1\}$, $i=1, \dots, N$, representing the sample points and outcomes, where

$$Y_i \sim \text{Bernoulli}(f(X_i)). \quad (1)$$

The success probability function, denoted by $f: \mathcal{B} \rightarrow [0, 1]$, represents the true probability of achieving the goal on a particular simulation trial. While this function is unknown in practice (the purpose of the Separatrix Algorithm is to estimate this function near the user-specified isocline), it is safe to assume that f is sufficiently regular in terms of continuity and differentiability.

Let M denote the number of sample points to be selected on each iteration, and note that M should be selected to maximally leverage parallel computing resources. These “next points” to sample will be selected by the Separatrix Algorithm so as to gather information about an isocline, denoted by f^* , of the underlying success probability function. We will refer to this isocline of the success probability function as the separatrix, see Figure 1.

The first samples are selected using a space filling algorithm, such as LHS. Then, the main iteration loop consists of two

primary steps. The *inference step* takes available samples, X , and outcomes, Y , as inputs, and returns a probability density function for the probability of success at one or more *inference points*. The *design of experiments step* uses the data to select subsequent sample points. Each of these algorithms will be described in detail in the following sections, and are summarized in Algorithm Summary S1.

Stochastic Inference using Kernel Regression

The inference portion of the Separatrix Algorithm uses a novel variation of kernel regression to estimate the *distribution* of the probability of success at a specified inference point, $X^* \in \mathcal{B}$. This differs from most kernel methods, which simply estimate a single value, representative of the mean, at each inference point.

The output from each simulation is passed through a Boolean test function that returns “true” only if the user-specified goal is achieved. Because each simulation is an independent Bernoulli trial, if one or more simulation are run with a *fixed* parameter configuration, the resulting probability of success after observing α successes and β failures would have a beta distribution,

$$f \sim \text{Beta}(\alpha + 1, \beta + 1). \tag{2}$$

However, it is inefficient to run multiple simulations with a fixed parameter configuration, and we often wish to compute the distribution at a point for which no outcomes have been observed.

Our approach to the inference problem at X^* is to first estimate the effective number of observed successes, $\hat{\alpha}(X^*)$, and failures, $\hat{\beta}(X^*)$, using kernel methods, and then plug these estimates into the beta distribution,

$$\hat{f}(X^*) \sim \text{Beta}(\hat{\alpha}(X^*) + 1, \hat{\beta}(X^*) + 1). \tag{3}$$

Kernel methods, commonly used in density estimation and regression, compute the result value as a (kernel) weighted average of “nearby” values. Nearness is determined by the spatial scale of the kernel, called the bandwidth. Many techniques have been proposed for selecting the kernel bandwidth, here we derive the kernel scale from the distance to the k^{th} nearest neighbor (KNN).

When the values are sampled uniformly, classical kernel methods are consistent, meaning that the estimate is asymptotically correct in probability. However, direct application of these concepts results in an unacceptably large bias when the data are not sampled uniformly. This bias in classical kernel methods is unacceptable for direct application in the Separatrix problem because the very objective of the algorithm is to concentrate samples near the separatrix, resulting in a non-uniform sampling density that becomes increasingly non-uniform as additional samples are collected.

To minimize bias with highly non-uniform sampling, the inference portion of the Separatrix Algorithm uses a density correction factor in the kernel. Bias correction has been used in binary kernel regression by Hazelton [50], however our approach differs. We proceed by describing the two main sub-steps of the inference portion of the Separatrix Algorithm: (1) estimate the sample point density using standard kernel methods, and (2) apply a density-corrected kernel to estimate the number of successes and failures, which then feeds into a beta distribution to finally approximate the distribution of the probability of success at each inference point.

Density Estimation. The density estimator uses standard variable kernel smoothing techniques [51] to produce an estimate of the sample point density at each sample point. While any consistent density estimator could be used, variable kernel methods are known to produce a bona fide density estimate that is globally consistent, however the edges converge at a slower rate [52].

The density estimate at point $x \in \mathcal{B}$ is computed as

$$\hat{\rho}(x) = \frac{1}{N} \sum_{j=1}^N \frac{1}{h_{\rho} d_{j,k_{NN,\rho}}} K\left(\frac{x - X_j}{h_{\rho} d_{j,k_{NN,\rho}}}\right). \tag{4}$$

Here, $d_{j,k_{NN,\rho}}$ is the distance from sample point j to the $k_{NN,\rho}$ -th nearest sample point. The variable kernel method employs a bandwidth that is proportional, with constant h , to this distance. Many authors have proposed methods to select $k_{NN,\rho}$ automatically; we use the parametric form

$$k_{NN,\rho}(N) = cN^{\gamma}, \tag{5}$$

for constants c and γ .

Due to extreme density variations driven by the desire to sample near the separatrix, we replace the global N in (5) with a local estimate of the number of sample points at each inference point, $\hat{N}(x)$, using a scale model,

$$\hat{N}(X^*) = \sum_{i=1}^N \exp\left(-\frac{(X_i - X^*)^2}{2\sigma_{\hat{N}}^2}\right), \tag{6}$$

in which $\sigma_{\hat{N}}$ is a fixed bandwidth. Note the lack of kernel normalization above. Instead, the kernel is one when $X_i = X^*$ so that a sample at X^* counts as one “effective” sample, but samples further from X^* count only partially in the total.

Success Probability Distribution Inference. With the density estimates available, the effective number of successes and failures at X^* are calculated as,

$$\hat{\alpha}(X^*) = \mu(X^*) \sum_{i=1}^N \left[\frac{L_g(X_i, X^*) Y_i}{\hat{\rho}(X_i)} \right] \tag{7}$$

$$\hat{\beta}(X^*) = \mu(X^*) \sum_{i=1}^N \left[\frac{L_g(X_i, X^*) (1 - Y_i)}{\hat{\rho}(X_i)} \right]. \tag{8}$$

Here, μ is a scaling factor (to be explained shortly), and $L_g : \mathcal{B}^2 \rightarrow \mathcal{R}^+$ is a kernel function with bandwidth g . We use a squared exponential kernel with a simple S-type (i.e. diagonal) bandwidth matrix, for which the scaling is determined by the distance to the $k_{NN,i}$ -th nearest neighbor, as in (5), although the parameters are not necessarily the same. Again, we replace the global N by $\hat{N}(x)$ from the scale model.

The scaling constant, denoted by μ , is selected so that the number of success and failure outcomes at X^* sums to the estimated “effective number of samples” at X^* from the scale model (6),

$$\mu(X^*) = \frac{\hat{N}(X^*)}{\sum_{i=1}^N \left[\frac{L_g(X_i, X^*)}{\hat{\rho}(X_i)} \right]} \tag{9}$$

This scaling by μ in the calculation of $\hat{\alpha}$ and $\hat{\beta}$ has no impact on mode of the beta distribution (8), but does affect higher order moments.

The distribution of the probability of achieving the goal at each test point is approximated by a beta distribution (3). The approximation comes from the fact that the true posterior has a beta distribution only for repeated simulations (i.e. fixed parameters).

Design of Subsequent Experiments

The second main step of the Separatrix Algorithm is a design-of-experiments (DOE) procedure that selects M new sample points in a manner that actively focuses computational effort on interesting regions of parameter space. This is counter to traditional factorial, space-filling, and variance-reducing approaches, which sample globally. Traditional Bayesian design-of-experiments (BDOE) methodology, see the original work [53] or a review [54], also places samples globally. However, samples in BDOE are actively placed so as to maximize the expected information gain, as measured by a Kullback-Liebler (KL) pseudometric. The experiment design portion of the Separatrix Algorithm uses an ‘‘interest guided’’ (igBDOE) approach in which the distributions used in the KL calculation are tailored in a novel way that encourages subsequent samples to focus on identifying the separatrix.

More specifically, the BDOE methodology chooses new input configurations, $X^+ \in \mathcal{B}^M$, that are expected to maximally distinguish the posterior distribution, which includes the M proposed samples, from the prior distribution, which does not:

$$X^+ = \arg \max_{\chi \in \mathcal{B}^M} E_{Y(\chi)} \left[\int_{\mathcal{B}} D_{KL}(\hat{f}(t | X, Y) \| \hat{f}(t | X, Y, \chi, Y(\chi))) dt \right] \tag{10}$$

Here, $t \in \mathcal{B}$ is a test point at which the prior and posterior distributions are to be compared, $\hat{f}(t | \dots)$ is the inferred success probability distribution at t , χ is the collection of M sample points in consideration, and $Y(\chi) \in \{0, 1\}^M$ is the corresponding outcome vector. In a small abuse of notation, $E_{Y(\chi)}$ denotes expectation with respect to the *distribution* over outcomes at χ , which in turn is calculated from the expected value of $\hat{f}(\chi | X, Y)$. Finally, D_{KL} denotes the Kullback-Liebler divergence between distributions v and w ,

$$D_{KL}(v \| w) = E_v \log(v/w) \tag{11}$$

Without modification, traditional Bayesian experiment design selects new inputs that are globally important.

In contrast, the *interest-guided BDOE* (igBDOE) algorithm used in the Separatrix Algorithm focuses subsequent samples on the regions of parameter space for which the success probability is near f^* . Rather than applying the traditional BDOE methodology to the beta distribution from the inference algorithm, igBDOE

applies a similar approach to a discretization we call the *interest distribution*,

$$q_0(t | X, Y) = \text{Prob}(\hat{f}(t | X, Y) < f^*) \tag{12}$$

$$q_1(t | X, Y) = 1 - q_0(t | X, Y) \tag{13}$$

Here $q_0(t | X, Y)$ is the total probability mass of distribution $f(t | X, Y)$ that lies below the interest level, and $q_1(t | X, Y)$ is the mass above the interest level. Similarly, $q_{\{0,1\}}(t | X, Y, \chi, Y(\chi))$ is the total probability mass of distribution $f(t | X, Y, \chi, Y(\chi))$ that lies below (above) the interest level. In this way, the experiment design procedure selects new inputs that are expected to move mass across separatrix. As a consequence, the procedure samples primarily in and around the desired isocline, as these locations have the greatest potential to change the interest distribution, and occasionally samples far away when uncertainty is sufficiently high.

Practically, Bayesian approaches to experiment design tend to be computationally intensive. To make the computation tractable, we apply a number of simplifications. In place of the integral in (10), we restrict consideration to T test points $t_j \in \mathcal{B}$, $j = 1, \dots, T$. Instead of choosing all M new sample points jointly, we choose M points that *independently* result in a large expected KL divergence. As a final simplification for speed and tractability, we optimize not over the entire parameter space ($\chi \in \mathcal{B}^M$), but rather select from a finite number $C > M$ of *candidate sample points*, $\chi \subset \{\xi_1, \xi_2, \dots, \xi_C\}$, $\xi_i \in \mathcal{B}$. With these simplifications, the igBDOE chooses M of C candidate points so as to maximize the sum of the expected (over outcomes) mean (over test points) KL divergence,

$$X^+ = \max_{\mathcal{I} \subset \{1, \dots, C\}} \sum_{i \in \mathcal{I}} J(\xi_i), \quad \text{subject to } |\mathcal{I}| = M \tag{14}$$

where

$$J(\xi_i) = E_{Y(\xi_i)} \left[\frac{1}{T} \sum_{j=1}^T D_{KL}(q(t_j | X, Y) \| q(t_j | X, \xi_i, Y(\xi_i))) \right] \tag{15}$$

Selection of the T test points and C candidate points can have a significant impact on the performance of the algorithm. A simple approach is to choose these points using a LHS design. However, the resolution of test and sample points quickly becomes the limiting step as the separatrix becomes increasingly refined. To increase the density of test and sample points in places where the KL divergence is high, we propose sampling test and candidate points from the variance of the interest distribution,

$$p(t) = q_0(t | X, Y) q_1(t | X, Y) \tag{16}$$

$$= q_0(t | X, Y) - q_0^2(t | X, Y) \tag{17}$$

In practice, these samples are obtained using Markov chain Monte Carlo (MCMC). An optional post-processing step slightly diffuses the points by adding a random value sampled from a narrow multivariate normal distribution, having diagonal standard deviation denoted by σ_b , to each point from MCMC. The number

of candidate points can also be varied, and in practice we find choosing $C = \lambda M$ with $\lambda \in [2, 5]$ works well for all applications. Finally, to prevent the algorithm from being overly greedy when the number of observed samples is small, we initially limit the number of subsequent sample points selected by igBDOE to a fraction, z , of the total number of samples, e.g. $M = \min(\text{floor}(zN), M)$. The remaining $M - M$ samples, if any, are obtained using LHS.

Results

To illustrate the main contributions of this work, we have prepared a variety of results that demonstrate the Separatrix Algorithm, which has been implemented in Matlab [55]. A basic implementation of the algorithm is available in Source Code S1. We begin with a simple one-dimensional problem to show the full distribution of the estimated success probability function, and also to clearly illustrate the experiment design process for generating subsequent sample points. The methods are then applied to a two-dimensional problem with a known test function that resembles those encountered in epidemiological decision making. Finally, an agent-based stochastic simulation model of malaria [56,57] is examined to show how the Separatrix Algorithm applies to a computer simulation having three separatrix dimensions. This simulation model was coded in C++ by a team of researchers and developers at the Institute for Disease Modeling in Bellevue, WA. Simulations were invoked on a supercomputer running Windows®

HPC Server with more than 5000 cores, however we choose M between 50 and 250 cores for these examples. The job submission pipeline and commissioning scripts use a mixture of Matlab, Python, and C#.

Example One: Hyperbolic Tangent in 1D

The basic concepts of the method can be seen most vividly on a simple one-dimensional toy problem. Denoting by x the independent parameter, the true probability of success for this example is a hyperbolic tangent,

$$f(x) = \frac{1}{2} (\tanh(10(x - 0.6)) + 1), \mathcal{B} = [0, 1]. \quad (18)$$

This test function is plotted in Figure 2A, along with outcomes observed at $N_0 = 50$ initial samples that were drawn using a LHS design. For display purposes, the posterior beta distribution (3) was evaluated at 50 equally spaced inference points, and is displayed in hypercolor along with the mode.

For this example, we have selected a target success probability of $f^* = 70\%$, for which the sinusoidal model crosses this target at a separatrix of $x = 0.642$. Additional samples were selected $M = 50$ at a time until a total of $N_f = 1000$ samples had been observed. At most $\text{floor}(0.25N)$ samples are added using the igBDOE algorithm

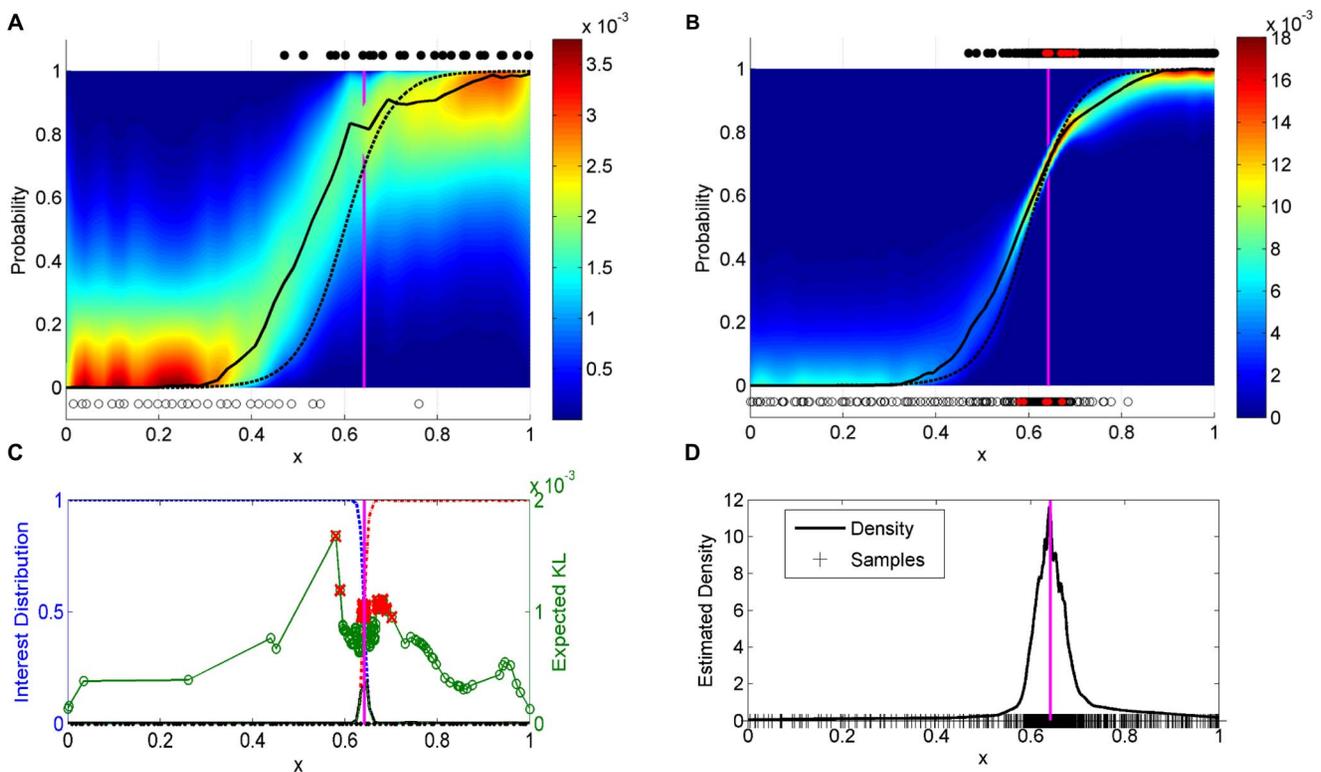


Figure 2. One-dimensional hyperbolic tangent analysis. (A) Shown are the true success probability function (dashed line), 50 LHS samples (full and empty circles), the inferred distribution (hypercolor), and the most likely value (black line). The vertical magenta line is at the separatrix corresponding to an interest level of $f^* = 0.7$. (B) The probability density after observing $N_f = 1000$ samples using the Separatrix Algorithm. Note that the estimate is tight near the separatrix. (C) The inner workings of the igBDOE algorithm. First, test and sample points are loaded from the previous iteration in which they were sampled from the variance of the interest distribution, solid black (left axis), which in turn is computed from the interest distribution: q_0 is in blue-dash and q_1 is in red dash-dot. The expected KL divergence is plotted for each of the candidate sample points (green circles, right axis). The best M of these candidates, indicated by red crosses, will be selected. (D) The final density estimate shows that the igBDOE algorithm was placing samples in and around the separatrix. Ticks on the x-axis represent samples. doi:10.1371/journal.pone.0103467.g002

Table 1. Parameters values.

Param	Meaning	Example 1	Example 2	Example 3
f^*	Interest level	0.7	0.6	0.5
N_0	Initial number of samples	50	50	250
M	New points per iteration	50	50	250
N_f	Final number of samples	1000	1500	2500
C	Number of candidates	200	200	1000
T	Number of test points	100	100	100
z	Max fraction from igBDOE	0.25	0.25	0.5
h_p	Density bandwidth scaling	0.3	0.3	0.3
$k_{NN,p}$	KNN for density	$0.5N^{0.8}$	$0.5N^{0.67}$	$0.5N^{0.57}$
$k_{NN,i}$	KNN for inference	$4\hat{N}^{0.8}$	$4\hat{N}^{0.67}$	$4\hat{N}^{0.57}$
$\sigma_{\hat{N}}$	Counting kernel scale for \hat{N}	0.02	0.05	0.07
σ_b	Blur kernel bandwidth	0.02	0.02	0.02

Separatrix parameter values used in the three example presented in this paper.
doi:10.1371/journal.pone.0103467.t001

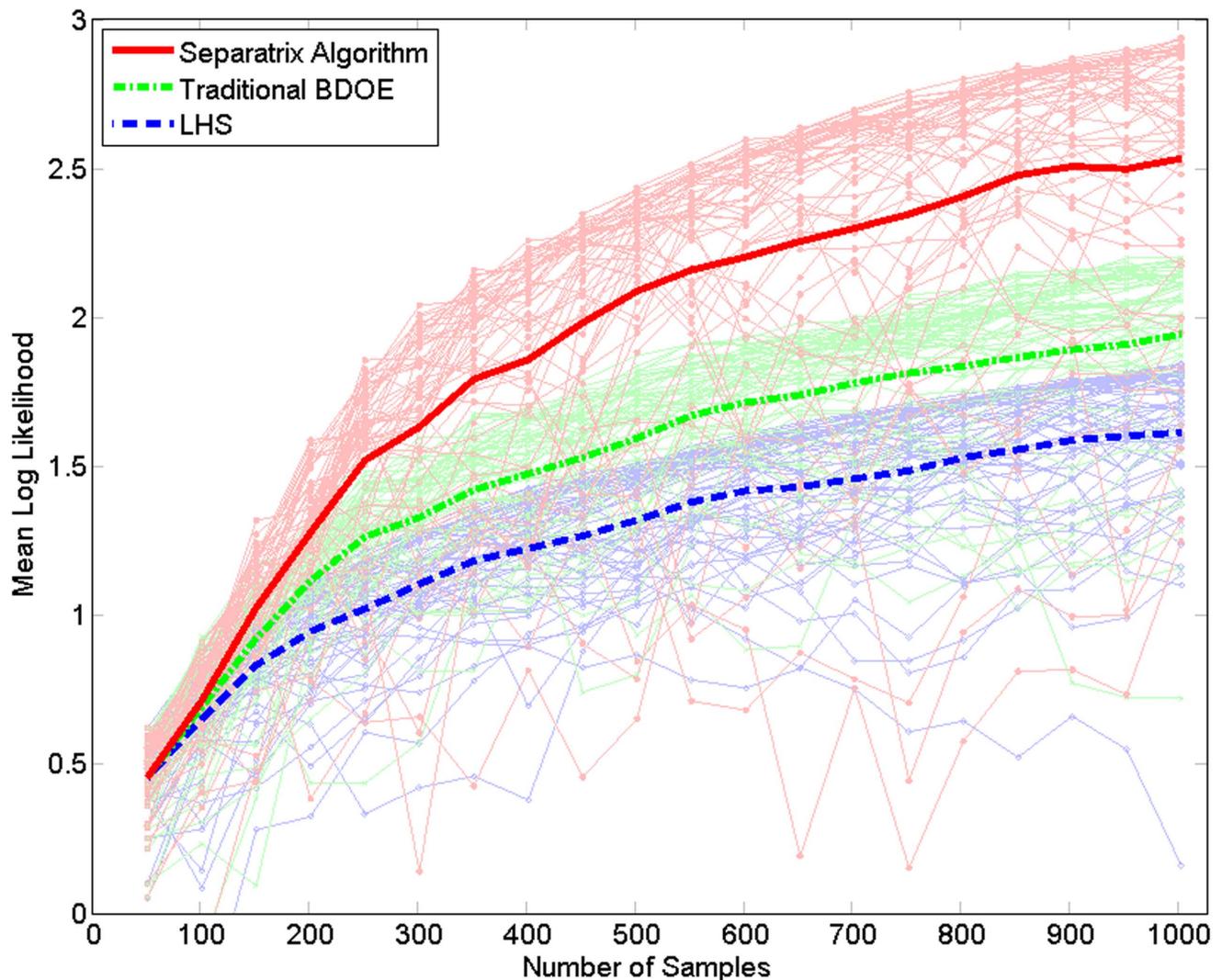


Figure 3. One-dimensional hyperbolic tangent performance. For the one-dimensional hyperbolic tangent test function (18), the Separatrix Algorithm outperforms Latin hypercube sampling and traditional BDOE on a likelihood-based performance metric (19).
doi:10.1371/journal.pone.0103467.g003

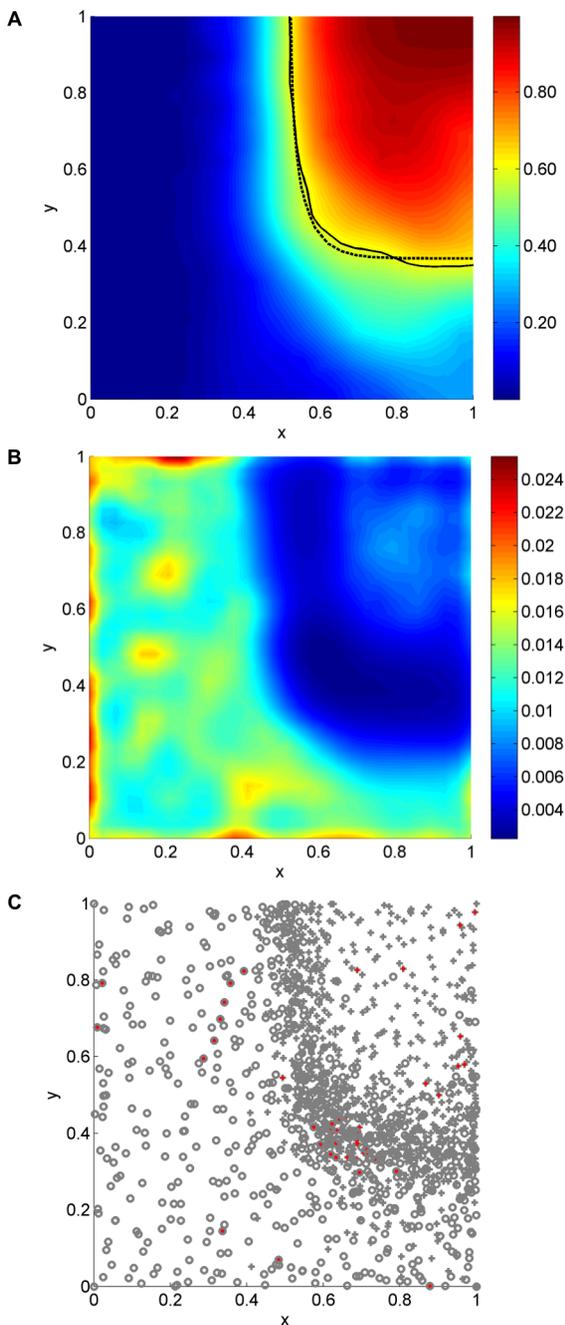


Figure 4. Two-dimensional separatrix results. The inference algorithm was applied at all points on a regular 30×30 grid after collecting $N_f = 1500$ samples. Here, we display the mode (A), variance (B), and samples (C). The dashed line in (A) is the true separatrix, and the solid line is the estimate. Circles and crosses in (C) represent failures and successes, respectively, and red dots indicate samples selected on the final iteration.
doi:10.1371/journal.pone.0103467.g004

per iteration, and remaining sample points come from LHS. A complete list of parameters is available in Table 1.

Upon termination, the probability density clearly reveals the location at which the test function crosses 70%, see Figure 2B. The interest-guided experiment design procedure has concentrated

samples at the separatrix, see the sample point density estimate in Figure 2D.

The inner workings of the igBDOE experiment design can be seen in Figure 2C, which shows selection of the final 50 samples. Here, the interest distribution is shown as a red dashed line for q_0 (12) and as a blue dashed-dot line for q_1 (13). The variance of the interest distribution (17), from which test and candidate points are sampled, is shown in black solid. The expected KL divergence (10) is evaluated at each candidate sample point, the green circles, and the best $M = 50$ candidates, indicated by the red crosses, will be simulated on the next iteration.

To compare the performance of the Separatrix Algorithm to Latin hypercube sampling and traditional BDOE, we evaluated a metric for 60 random number seeds of each algorithm. In particular, we computed the mean log likelihood at N_e evaluation point(s), X_{e_i} , on the separatrix,

$$L = \frac{1}{N_e} \sum_{i=1}^{N_e} \log \left(\frac{(f^*)^{\hat{\alpha}(X_{e_i})} (1-f^*)^{\hat{\beta}(X_{e_i})}}{B(\hat{\alpha}(X_{e_i}) + 1, \hat{\beta}(X_{e_i}) + 1)} \right). \quad (19)$$

The function denoted by B in the denominator of the log is the beta function. For this one-dimensional example, we used a single evaluation point on the $f^* = 0.7$ separatrix, $X_{e_1} = 0.642$. The Separatrix Algorithm outperforms LHS and traditional BDOE, as can be seen in Figure 3.

Example Two: Hyperbolic Tangent in 2D

To extend the analysis to a two-dimensional example, we consider a test function based on a product of two hyperbolic tangents,

$$f(x,y) = 1/4(\tanh(10x - 5) + 1)(\tanh(3y - 0.9) + 1), \quad (20)$$

where $\mathcal{B} = [0,1] \times [0,1]$.

This test function resembles certain epidemiological systems in that the probability of success is low without the interventions (0,0) and high with both interventions at full coverage (1,1) and that the probability surface can have different steepness and critical coverage for different interventions.

We configured the interest-guided experiment design procedure to select the best $M = 50$ points from $C = 200$ candidates, and terminated when $N_f = 1500$. Parameters were generally similar to those used in the previous example, see Table 1.

The final estimate from a typical separatrix analysis is shown in Figure 4. As with the one-dimensional example, the fit is not intended to be good away from the separatrix. In fact, the interest-guided experiment design has again done well to place samples in and around the separatrix, see Figure 4C. The separatrix is more difficult to identify along the horizontal portion than the vertical portion due to the slope. The Separatrix Algorithm has correspondingly placed more samples along the horizontal portion of the separatrix, resulting in reduced variance there.

We have again compared the Separatrix Algorithm to a LHS experiment design, using the mean log-likelihood metric (19). This metric naturally balances accuracy and precision, and was evaluated at $N_e = 20$ points spaced equally in arc-length along the separatrix. The results show that the Separatrix Algorithm significantly and consistently outperforms LHS, see Figure 5.

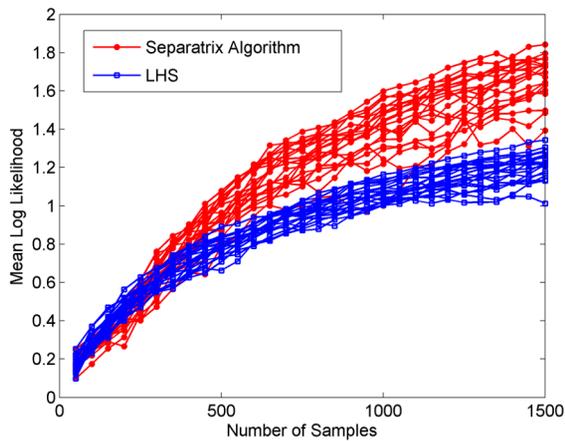


Figure 5. Two-dimensional hyperbolic tangent performance. The Separatrix Algorithm again outperforms Latin hypercube sampling on the mean log likelihood metric, which was evaluated at 20 points spaced evenly in arc-length along the separatrix. doi:10.1371/journal.pone.0103467.g005

Example Three: Individual-Based Malaria Simulation

In epidemiology, three-dimensional parameter sweeps are often conducted to explore trade-offs between competing alternatives in the context of a third parameter. We now apply the Separatrix Algorithm to a detailed computer simulation of malaria [56] for which the true response function is unknown.

This separatrix analysis considers the goal of local malaria elimination using an intervention composed of insecticide treated bednets (ITN) and a pre-erythrocytic vaccine (PEV). Each individual-based computer simulation returns true if and only if malaria is eliminated from the model within 10 years. The success of such a campaign will depend on the local burden of disease and on the extent to which mosquitoes feed indoors. The malarial disease burden is typically quantified by the entomological inoculation rate (EIR), which quantifies the number of infectious bites received by a typical person in one year. The higher the EIR, the more difficult elimination will be. ITNs are well characterized, here we assume that bednets are distributed to 80% of the population, block 90% of indoor bites, and kill 60% of the mosquitoes attempting indoor feeds. Naturally, bednets will not be as useful of an intervention against predominantly outdoor-feeding mosquitoes.

It is in this context that we apply the Separatrix Algorithm to explore the potential impact of a PEV, when used in tandem with ITN. Vaccines against malaria are currently under development, so it is not clear what efficacy and durability to expect. Here, we have assumed a vaccine efficacy against infection that decays exponentially with a half-life of 3.5 years, and use separatrix analysis to explore *initial* efficacy.

We initially selected $N_0 = 500$ samples points using a LHS design. The Separatrix Algorithm then iteratively selected $M = 250$ samples from $C = 1000$ candidates. As with the previous examples, igBDOE was initially complimented by LHS. The algorithm was terminated once $N_f = 2500$ simulations had been conducted. Parameter values can be found in Table 1. The resulting separatrix plot, variance, and sample points are displayed in Figure 6.

From the separatrix plot, local elimination is possible in simulated scenario using ITN alone (PEV efficacy of zero) if the intensity of malaria transmission (EIR) is sufficiently low and

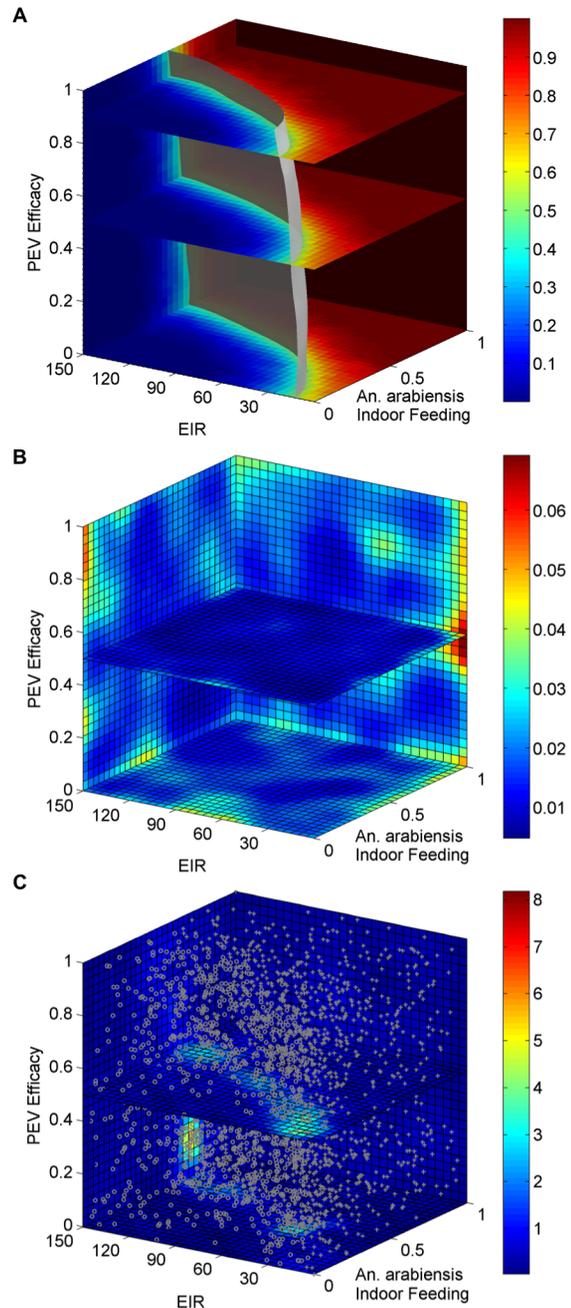


Figure 6. Malaria model separatrix results. The separatrix (A), variance (B), and samples with density (C) after simulating the malaria model $N_f = 2500$ times. doi:10.1371/journal.pone.0103467.g006

mosquitoes tend to feed indoors where bed nets are effective [58]. Increasing vaccine efficacy expands the region of EIR/mosquito behavior space in which elimination is achievable.

Discussion

To effectively leverage a stochastic simulation model in the goal-driven decision making process, we have presented the Separatrix Algorithm. The tool has potential uses in a multitude of domains,

and has been discussed here in the context of infectious disease policy making. The algorithm proceeds iteratively wherein each iteration combines binary kernel regression with an interest-guided experiment design. This procedure was demonstrated on three example problems in which the Separatrix Algorithm was able to clearly isolate combinations of intervention and model parameters for which the goal is achieved with a user-specified probability of success.

In order to apply the Separatrix Algorithm, a few preliminary requirements must be satisfied. These consist primarily of a stochastic simulation with externally configurable parameters, user-specified Boolean function measuring the success of each simulation, and a target success probability isocline. Additionally, the user is required to select the region of parameter space to be explored, and it assumed that parameters not explored by the algorithm have been calibrated externally. The parameter space, \mathcal{B} , need not come directly from the input space, but instead could be mapped through a coordinate transformation to address parametric synergies. For example in malaria, immune response thresholds to within-host parasite densities and detection thresholds for within-host parasite densities can be moved together without substantially affecting the proportion detected, so some parameter explorations may choose a single dimension for both parameters. Also, the ability of the local vector population to transmit can be represented at a given point in time by a single quantity called the vectorial capacity [59].

The Separatrix Algorithm has several important limitations that should be considered. The problem being addressed by the algorithm is inherently difficult, and the complexity grows combinatorially with the number of dimensions. Therefore, separatrix exploration of a large parameter space (e.g. $D > 6$, see [60]) requires access to a powerful workstation to run the algorithm and a supercomputer to run the stochastic simulations. With increasing dimensions, the number of sample points (e.g. simulations) required to achieve a specified level of confidence in the location of the separatrix increases. The computational complexity of the Separatrix Algorithm, neglecting the optional Markov chain Monte Carlo step, scales as $O((T + C)N^2)$. However much of the repeated computation can be cached.

The algorithm has several user-configurable parameters, including the initial number of samples, the number of samples to add on each iteration, kernel bandwidth scale factors, and so on. These parameters are inherently problem specific, and yet all three examples above were computed using similar parameters, see Table 1. Also, some analyses will be more challenging than others, depending in part on the slope of the success probability function at the separatrix. Further work is needed to determine optimal parameter values for a given problem, and automatic parameter selection could increase algorithm performance. Until an automatic procedure for selecting parameter values is available, a technician familiar with the algorithm, parameters, and implementation may be required to assist the decision maker in conducting the analysis.

More time-consuming kernel methods use complex bias reduction techniques based on pilot estimates, and cross-validation or other optimization procedures for bandwidth selection. However, the separatrix context differs from other applications because additional data can be collected. Time spent perfecting the inference or experiment design is time that could otherwise be used to generate new outcomes. For a simulation that takes days to run, it may be worthwhile to use more advanced, and computationally-intensive, kernel methods. However, for simulations taking on the order of minutes to hours, the methods we have

outlined above provide a balance between experiment design and simulation.

Conclusions

We have presented the Separatrix Algorithm for evaluating a detailed stochastic simulation with respect to a goal-oriented question and user-specified probability of success. The method combines bias-corrected kernel smoothing with an interest-guided experiment design to efficiently focus computational resources on identifying the desired separatrix. The method was applied to several toy problems representing typical response functions and to a stochastic epidemiological simulation of malaria.

Additional layers of model analysis can be included once the separatrix has been located by the Separatrix Algorithm. For example, if costing and resource allocation models are available, they can be evaluated and optimized within the separatrix. Additionally, parameters within the separatrix can be preferentially selected based on a prior or posterior parameter distribution.

The method could be improved in several ways. Numerous advances in multivariate kernel techniques have been published in the past few years. We have primarily implemented well-established techniques, but newer algorithms and automatic bandwidth selection procedures could improve the quality of separatrix plots at the cost of additional computing time. Additionally, we did not implement any correction at the edges of the parameter space. Higher-order kernels and local linear models are known to perform better at the edges.

There is no way to avoid the fact that parameter space exploration requires a considerable amount of computing power. Users lacking access to a large-scale computer should consider cloud computing, running smaller iterations, or reducing the number of dimensions. In any case, the Separatrix Algorithm uses fewer samples to achieve a desired level of precision, thereby saving time and/or money.

The experiment design procedure selects M points independently. A significant performance benefit could be achieved if these points could be selected jointly, even if this means selecting only two at a time.

Finally, epidemiological models often exhibit monotonic responses. A powerful addition to the Separatrix Algorithm would be to enable user specification or automatic detection of monotonic dimensions. The approach could be based on recent advances in monotonic kernel methods [61].

Supporting Information

Algorithm Summary S1 Algorithm Summary S1 contains a mathematical description of the Separatrix Algorithm. (PDF)

Source Code S1 Source Code S1 contains demonstration code written in Matlab. (ZIP)

Acknowledgments

The authors thank Bill and Melinda Gates for their active support of this work and their sponsorship through the Global Good Fund. This work was performed at Intellectual Ventures Laboratories with help from colleagues, especially Edward Wenger and Matthew Behrend.

Author Contributions

Conceived and designed the experiments: DJK MB PE. Performed the experiments: DJK. Analyzed the data: DJK. Wrote the paper: DJK MB PE.

References

1. Alonso P, Brown G, Arevalo-Herrera M, Binka F, Chitnis C, et al. (2011) A research agenda to underpin malaria eradication. *PLoS medicine* 8: e1000406.
2. Henderson R (1989) The world health organization's plan of action for global eradication of poliomyelitis by the year 2000a. *Annals of the New York Academy of Sciences* 569: 69–85.
3. Sabin A, Ramos-Alvarez M, Alvarez-Amezquita J, Pelon W, Michaels R, et al. (1960) Live, orally given poliovirus vaccine. *JAMA: the journal of the American Medical Association* 173: 1521.
4. Salk J (1955) Considerations in the preparation and use of poliomyelitis virus vaccine. *Journal of the American Medical Association* 158: 1239.
5. Grassly N, Fraser C, Wenger J, Deshpande J, Sutter R, et al. (2006) New strategies for the elimination of polio from india. *Science* 314: 1150.
6. Stover J (2004) Projecting the demographic consequences of adult hiv prevalence trends: the spectrum projection package. *Sexually transmitted infections* 80: i14.
7. Stover J (2011) Hiv models to inform health policy. *Current Opinion in HIV and AIDS* 6: 108.
8. Fraser C, Donnelly C, Cauchemez S, Hanage W, Van Kerkhove M, et al. (2009) Pandemic potential of a strain of influenza a (h1n1): early findings. *Science* 324: 1557.
9. Brown S, Tai J, Bailey R, Cooley P, Wheaton W, et al. (2011) Would school closure for the 2009 h1n1 influenza epidemic have been worth the cost?: a computational simulation of pennsylvania. *BMC public health* 11: 353.
10. Ferguson N, Cummings D, Cauchemez S, Fraser C, Riley S, et al. (2005) Strategies for containing an emerging influenza pandemic in southeast asia. *Nature* 437: 209–214.
11. Eubank S, Guclu H, Kumar V, Marathe M, Srinivasan A, et al. (2004) Modelling disease outbreaks in realistic urban social networks. *Nature* 429: 180–184.
12. Thompson K, Tebbens R (2008) Using system dynamics to develop policies that matter: global management of poliomyelitis and beyond. *System Dynamics Review* 24: 433–449.
13. Breman J, Allio M, Mills A (2004) Conquering the intolerable burden of malaria: what's new, what's needed: a summary. *The American journal of tropical medicine and hygiene* 71: 1–15.
14. Smith T, Maire N, Ross A, Penny M, Chitnis N, et al. (2008) Towards a comprehensive simulation model of malaria epidemiology and control. *Parasitology* 135: 1507–1516.
15. Smith T, Killeen G, Maire N, Ross A, Molineaux L, et al. (2006) Mathematical modeling of the impact of malaria vaccines on the clinical epidemiology and natural history of plasmodium falciparum malaria: Overview. *The American journal of tropical medicine and hygiene* 75: 1–10.
16. Griffin J, Hollingsworth T, Okell L, Churcher T, White M, et al. (2010) Reducing plasmodium falciparum malaria transmission in africa: a model-based evaluation of intervention strategies. *PLoS medicine* 7: e1000324.
17. McKenzie F, Baird J, Beier J, Lal A, Bossert W (2002) A biologic basis for integrated malaria control. *The American journal of tropical medicine and hygiene* 67: 571–577.
18. Helton J, Davis F (2003) Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliability Engineering & System Safety* 81: 23–69.
19. McKay M, Beckman R, Conover W (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 23: 239–245.
20. Marino S, Hogue I, Ray C, Kirschner D (2008) A methodology for performing global uncertainty and sensitivity analysis in systems biology. *Journal of theoretical biology* 254: 178–196.
21. Castiglione F, Pappalardo F, Bernaschi M, Motta S (2007) Optimization of haart with genetic algorithms and agent-based models of hiv infection. *Bioinformatics* 23: 3350.
22. Lollini P, Motta S, Pappalardo F (2006) Discovery of cancer vaccination protocols with a genetic algorithm driving an agent based simulator. *BMC bioinformatics* 7: 352.
23. Pappalardo F, Mastriani E, Lollini P, Motta S (2006) Genetic algorithm against cancer. *Fuzzy Logic and Applications*: 223–228.
24. Baesler F, Sepúlveda J (2001) Multi-objective simulation optimization for a cancer treatment center. In: *Simulation Conference, 2001. Proceedings of the Winter. IEEE, volume 2, pp. 1405–1411.*
25. Patel R, Longini I, Elizabeth Halloran M, et al. (2005) Finding optimal vaccination strategies for pandemic influenza using genetic algorithms. *Journal of Theoretical Biology* 234: 201–212.
26. Khuri A, Mukhopadhyay S (2010) Response surface methodology. *Wiley Interdisciplinary Reviews: Computational Statistics* 2: 128–149.
27. Kleijnen J (2009) Kriging metamodeling in simulation: A review. *European Journal of Operational Research* 192: 707–716.
28. Tekin E, Sabuncuoglu I (2004) Simulation optimization: A comprehensive review on theory and applications. *IIE Transactions* 36: 1067–1081.
29. Barton R (1998) Simulation metamodels. In: *Simulation Conference Proceedings, 1998. Winter. IEEE, volume 1, pp. 167–174.*
30. Barton R, Meckesheimer M (2006) Metamodel-based simulation optimization. *Handbooks in operations research and management science* 13: 535–574.
31. Blanning R (1975) The construction and implementation of metamodels. *simulation* 24: 177.
32. Ankenman B, Nelson B, Staum J (2008) Stochastic kriging for simulation metamodeling. In: *Proceedings of the 40th Conference on Winter Simulation. Winter Simulation Conference, pp. 362–370.*
33. Box G, Wilson K (1951) On the experimental attainment of optimum conditions. *Journal of the royal statistical society series b (methodological)* 13: 1–45.
34. Fang H, Horstmeier M (2006) Global response approximation with radial basis functions. *Engineering Optimization* 38: 407–424.
35. Sabuncuoglu I, Touhami S (2002) Simulation metamodeling with neural networks: an experimental investigation. *International Journal of Production Research* 40: 2483–2505.
36. Wahba G (1990) Spline models for observational data, volume 59. *Society for Industrial Mathematics.*
37. Kleijnen J, Van Beers W (2004) Application-driven sequential designs for simulation experiments: Kriging metamodeling. *Journal of the Operational Research Society*: 876–883.
38. Van Beers W, Kleijnen J (2004) Kriging interpolation in simulation: a survey. In: *Simulation Conference, 2004. Proceedings of the 2004 Winter. IEEE, volume 1.*
39. Morris M (1991) Factorial sampling plans for preliminary computational experiments. *Technometrics*: 161–174.
40. Jourdan A, Franco J (2010) Optimal latin hypercube designs for the kullback-leibler criterion. *ASTA Advances in Statistical Analysis* 94: 341–351.
41. Sacks J, Welch W, Mitchell T, Wynn H (1989) Design and analysis of computer experiments. *Statistical science* 4: 409–423.
42. Jin R, Chen W, Sudjianto A (2005) An efficient algorithm for constructing optimal design of computer experiments. *Journal of Statistical Planning and Inference* 134: 268–287.
43. Donohue J (1994) Experimental designs for simulation. In: *Simulation Conference Proceedings, 1994. Winter. IEEE, pp. 200–206.*
44. Kleijnen J (2005) An overview of the design and analysis of simulation experiments for sensitivity analysis. *European Journal of Operational Research* 164: 287–300.
45. De Groot M (1970) 1970. optimal statistical decisions.
46. Chick S (2006) Bayesian ideas and discrete event simulation: why, what and how. In: *Simulation Conference, 2006. WSC 06. Proceedings of the Winter. IEEE, pp. 96–106.*
47. Bernardo J (1979) Expected information as expected utility. *The Annals of Statistics* 7: 686–690.
48. Smith A, Verdinelli I (1980) A note on bayes designs for inference using a hierarchical linear model. *Biometrika*: 613–619.
49. Chick S, Ng S (2002) Simulation input analysis: joint criterion for factor identification and parameter estimation. In: *Proceedings of the 34th conference on Winter simulation: exploring new frontiers. Winter Simulation Conference, pp. 400–406.*
50. Hazelton M (2007) Bias reduction in kernel binary regression. *Computational statistics & data analysis* 51: 4393–4402.
51. Abramson I (1982) On bandwidth variation in kernel estimates—a square root law. *The Annals of Statistics*: 1217–1223.
52. Hall P, Wehrly T (1991) A geometrical method for removing edge effects from kernel-type non-parametric regression estimators. *Journal of the American Statistical Association*: 665–672.
53. Lindley D (1956) On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*: 986–1005.
54. Chaloner K, Verdinelli I (1995) Bayesian experimental design: A review. *Statistical Science*: 273–304.
55. MATLAB (2012) version 8.1 (R2013a). Natick, Massachusetts: The MathWorks Inc.
56. Eckhoff P (2011) A malaria transmission-directed model of mosquito life cycle and ecology. *Malaria Journal* 10: 303.
57. Eckhoff P (2013) Mathematical models of within-host and transmission dynamics to determine effects of malaria interventions in a variety of transmission settings. *The American Journal of Tropical Medicine and Hygiene* 88: 817–827.
58. Wenger E, Eckhoff P (2013) A mathematical model of the impact of present and future malaria vaccines. *Malaria Journal* 12: 126.
59. Saul A, Graves P, Kay B (1990) A cyclical feeding model for pathogen transmission and its application to determine vectorial capacity from vector infection rates. *Journal of applied ecology*: 123–133.
60. Scott D, Sain S (2005) Multidimensional density estimation. *Handbook of Statistics* 24: 229–261.
61. Hall P, Huang L (2001) Nonparametric kernel regression subject to monotonicity constraints. *The Annals of Statistics* 29: 624–647.